REMARKS

No claims have been added or cancelled. Claims 1-42 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 102(e) Rejection:

The Examiner rejected claims 1-42 under 35 U.S.C. § 102(e) as being anticipated by Aridor et al. (U.S. Patent 6,618,737) (hereinafter "Aridor"). Applicants respectfully traverse this rejection for at least the reasons below.

First of all Applicants note that the cited art, Aridor, has very little relevance to Applicants' invention as claimed. Aridor teaches a method for caching of read-only or non-mutable data in a clustered implementation of a Java Virtual Machine. In contrast, Applicants' claims pertain to the synchronization of application server session data in a distributed system. As will be discussed in more detailed below regarding the rejections of individual claims, Aridor's teachings do not disclose the subject matter of Applicants' claims. In fact, Aridor actually teaches away from specific limitations of Applicants' claims.

Regarding claim 1, contrary to the Examiner's assertion, Aridor fails to disclose a distributed store comprising a primary state of session data configured for access by a plurality of application servers. Aridor fails to mention anything regarding session data or a plurality of application servers. The meanings of the terms "session data" and "application servers" are well known in the art. The teachings of Aridor have nothing to do with either session data or application servers. The Examiner refers to Aridor's master node and cites column 9, lines 25-30. However, the cited passage does not refer to session data or a plurality of application servers. In contrast, the cited passage includes definitions of terms, such as "master node, "master object" and "proxy object". In Aridor's system, the objects of a Java application are distributed among a plurality processing nodes according to a clustered Java Virtual Machine implementation.

As taught by Aridor, a master node contains the master version of an object, where the master version of the object is the only version of the object on which modifications may occur. However, Aridor does not describe anything regarding a distributed store including a primary state of session data configured for access by a plurality of application servers. Aridor does not mention session data or application servers at all.

Additionally in regard to claim 1, Aridor fails to disclose a system configured to compare a client state of the session data to a benchmark of the client state to determine a subset of the attributes that have been modified in the client state. The Examiner cites column 10, lines 39-55 and refers to Aridor's micro-benchmarks. However, Aridor's micro-benchmarks are not benchmarks of a client state of session data that is compared to the client state of session to determine a subset of attributes that have been modified. Instead, Aridor uses the term benchmark in a completely different manner. Aridor describes the use of small, specialized testing and measurement programs used to gather statistics regarding the various implementations of Aridor's caching methods. Aridor teaches that benchmarks include "a tight loop in which they perform the relevant Java operation" and that "[t]he total amount of time to execute the loop is measured and divided by the number of iterations to get the amortized cost per operation" (Aridor, column 10, lines 41-42). Thus, Aridor is using a benchmark testing program to gather information and statistics regarding his system. For example, Aridor presents TABLE 1 (top of column 11) that lists the respective amortized cost for various Java operations as measured by Aridor's micro-benchmark program. The benchmark recited in Applicants' claim 1 is of a client state of session data that is compared to the client state of session to determine a subset of attributes that have been modified, not a testing program like the benchmark in Aridor. Clearly, Aridor does not disclose comparing a client state to a benchmark of the client state to determine a subset of the attributes that have been modified in the client state.

Moreover, Aridor's system does not involve any sort of comparison between the master object, which the Examiner presumably equates the primary state of Applicants' claim, and a cached object, which the Examiner presumably equates to

the client state of Applicants' claim. Instead, Aridor specifically teaches that only read-only data is cached. In some embodiments, Aridor teaches that data that may or may be read-only may be cached until the data is modified, as which time all caches of the data are invalidated. Aridor repeated stresses that modifications to data only occur at the master object on the master node. In fact, Aridor provides lengthy explanations regarding ensuring that the invalidation of all caches for a particular field occurs *prior* to updating the value of field in the master object on the master node. In Aridor's preferred embodiment, once a field is modified, the field is no longer cached and all accesses and updates are communicated to the master node for execution. In an alternative embodiment, after invalidation, the new value of a field may again be cached, but any subsequent modification again invalidates all caches of the field. Please see the following sections of Aridor: column 10, lines 1-2 and lines 15-23; column 11, lines 18 – 27 and lines 32-35; column 12, lines 59-65; column 15, lines 18-25 and lines 53-60; column 17, lines 24-32; column 18, lines 17-30; column 19, lines 33-39; column 24, lines 52-62; and column 25, lines 5-10.

Aridor's system simply does not involves modifying the value any cached field on nodes other than the master node and ensures that *all* cached versions of a modified field are invalidated prior to modifying the field on the master node. Thus, Aridor's system specifically does not involve comparing a client state to a benchmark of the client state to determine a subset of attributes that have been modified in the client state, since Aridor's system specifically guarantees that no attributes can be modified on cached versions of data. **Thus, not only does Aridor fail to disclose, Aridor actually teaches away from comparing the client state to a benchmark of the client state to determine a subset of the attributes that have been modified in the client state.**

Further in regard to claim 1, Aridor fails to disclose a system configured to synchronize the primary state with the client state according to the subset of the attributes. As described above, Aridor's system specifically prevents the modification of cached versions of data from being updated by invalidated all cached version of a modified field and only updates the field in the master object on the master node. All

subsequent accesses to the modified field are remote access implemented by communicated with the master node. Thus, Aridor's system does not, by design and desire, include synchronizing a primary state with a client state according to a subset of attributes that have been modified in the client state.

Applicants remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Aridor clearly fails to disclose numerous specific features recited in claim 1. In fact, as shown above, Aridor teaches away from specific limitations of claim 1. Therefore, Aridor cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 10, 19, 27 and 35.

Regarding claim 3, Aridor fails to disclose a system configured to perform binary differencing of a binary representation of the client state and a binary representation of the benchmark of the client state to locate the modified attributes. The Examiner cites column 12, lines 14-21 and refers to "4-byte words". However, the cited passage is not describing anything about comparing a client state to a benchmark of the client state or about performing binary differencing. In contrast, the cited passage describes the fact that the "Java memory model guarantees atomicity for modifications at the granularity of 4–byte words." Thus, the cited passage pertains to at what granularity the Java memory model ensures that operations are performed atomically (i.e. without the possible of interruption by another operation, thread, or process). The cited passage has absolutely no relevance to comparing a client state to a benchmark of the client state or to performing binary differencing of binary representations.

Furthermore, as described above regarding claim 1, Aridor's system is specifically designed to avoid modifying cached data fields, which the Examiner equates to the client state of Applicants' claim. Thus, for at least the reasons above, the rejection of claim 3 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 8, 12, 17, 20, 25, 28, 33, 36 and 39.

Regarding claim 4, Aridor fails to disclose a system configured to perform object graph differencing of an object graph representation of the client state and an object graph representation of the benchmark of the client state. The Examiner cites a particularly irrelevant passage (column 4, line 1) where Aridor describes smart proxies for objects. At the cited passage, Aridor teaches that given two array objects with different run-time behavior, a caching proxy may boost performance for a final static array (e.g. an array for which all access are read-only) and that remote access (i.e. no caching) should be used for an array that involves both read and write operations. Aridor does not teach or even mention anything regarding performing object graph differencing at the cited passage or anywhere else. The rejection of claim 4 is clearly not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 9, 13, 18, 21, 26, 29, 34, 37 and 42.

Regarding claim 5, Aridor fails to disclose a system configured to compare the tracked accessed attributes to a benchmark of the attributes of the client state to determine a subset of the tracked accessed attributes that have been modified in the client state and synchronize the primary state with the client state according to the subset of the tracked access attributes. The Examiner cites column 25, lines 48-63. However, the cited passage has no relevance to, and makes no mention of, comparing tracked accessed attributes to a benchmark of the attributes of the client state. Furthermore, as described above regarding the rejection of claim 1, this cited passage also fails to disclose anything regarding synchronizing a master version of a data field, which the Examiner equates to the primary state of Applicants' claim, with a cached version of a data field, which the Examiner equates to the client state of Applicants' claim. Please see the discussion of

claim 1 above for a more detailed discussion regarding the fact that not only does Aridor fail to disclose synchronization, Aridor actual teaches away from synchronizing.

Thus, for at least the reasons above, the rejection of claim 5 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 14, 22, 30, and 38.

Applicants also assert that the rejection of numerous other ones of the dependent claims is further unsupported by the cited art. However, since the rejection has been shown to be unsupported for the independent claims, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-12000/RCK.

Also enclosed herewith are the following items:

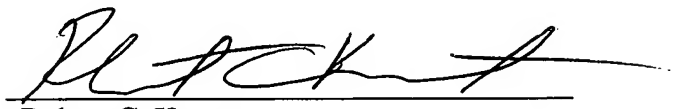
☒ Return Receipt Postcard

☐ Petition for Extension of Time

☐ Notice of Change of Address

☐ Other:

Respectfully submitted,



Robert C. Kowert
Reg. No. 39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: May 1, 2006